

## Association for Information Systems AIS Electronic Library (AISeL)

---

Research Papers

ECIS 2017 Proceedings

---

Spring 6-10-2017

# SEAMLESS UPDATES – HOW SECURITY AND FEATURE UPDATE DELIVERY STRATEGIES AFFECT CONTINUANCE INTENTIONS WITH DIGITAL APPLICATIONS

Tillmann Grupp

*Technische Universität Darmstadt, Darmstadt, Germany, [grupp@ise.tu-darmstadt.de](mailto:grupp@ise.tu-darmstadt.de)*

David Schneider

*Technische Universität Darmstadt, Darmstadt, Germany, [schneider@ise.tu-darmstadt.de](mailto:schneider@ise.tu-darmstadt.de)*

Follow this and additional works at: [http://aisel.aisnet.org/ecis2017\\_rp](http://aisel.aisnet.org/ecis2017_rp)

---

### Recommended Citation

Grupp, Tillmann and Schneider, David, (2017). "SEAMLESS UPDATES – HOW SECURITY AND FEATURE UPDATE DELIVERY STRATEGIES AFFECT CONTINUANCE INTENTIONS WITH DIGITAL APPLICATIONS". In Proceedings of the 25th European Conference on Information Systems (ECIS), Guimarães, Portugal, June 5-10, 2017 (pp. 611-626). ISBN 978-989-20-7655-3 Research Papers.  
[http://aisel.aisnet.org/ecis2017\\_rp/40](http://aisel.aisnet.org/ecis2017_rp/40)

This material is brought to you by the ECIS 2017 Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in Research Papers by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# SEAMLESS UPDATES – HOW SECURITY AND FEATURE UPDATE DELIVERY STRATEGIES AFFECT CONTINUANCE INTENTIONS WITH DIGITAL APPLICATIONS

*Research paper*

Grupp, Tillmann, Technische Universität Darmstadt, Darmstadt, Germany, grupp@ise.tu-darmstadt.de

Schneider, David, Technische Universität Darmstadt, Darmstadt, Germany, schneider@ise.tu-darmstadt.de

## Abstract

*Although updates have become the rule rather than the exception in modern digital ecosystems, to date they have received little attention in the IS post-adoption literature. We therefore draw on the IS continuance literature and expectation-confirmation theory to investigate, how different delivery strategies of security and feature updates impact users' continuance intentions (CI). Based on an online-experiment with 282 participants, we find a positive effect of security updates on users' CI only if users are notified after successful implementation. Feature updates, in contrast, elicit a positive effect on users' CI if they are at least announced before or after successful implementation. We also find that this positive effect of ex-ante announced feature updates diminishes if users have the choice to consume the update or not. In essence, our findings contribute to IS research by extending the mostly monolithic view of information systems by showing how an alterable information system might influence users' attitudes and behaviors during use. For practitioners, we show that it seems to be beneficial to inform users about updates, even though a silent integration has become possible with modern digital ecosystems, and that updates should be applied consistently. Directions for further research are discussed.*

*Keywords: Feature Updates, Security Updates, Delivery Strategies, IS continuance, IS post-adoption, expectation-confirmation theory*

## 1 Introduction

In most modern software ecosystems, where updates have become the rule rather than the exception, providers have strived for making the update process as integrated and unobtrusive as possible. Recently, with its newest release of the multi-device platform Android, Google has even announced to introduce 'seamless updates' (Samat, 2016). This is an update strategy, where updates are downloaded and installed completely in the background, without affecting application usage. Software updates, in this context, are no discrete and standalone programs themselves but are rather integrated into the base software to modify, extend or alter it, once they are applied to it (e.g., Dunn, 2004). From a user's perspective, two major update types delivering either additional functionality or security enhancements may be distinguished. Feature updates deliver additional functionality that extends the software with respect to its core purpose and are thus noticeable by users. Security updates remove potential vulnerabilities or enhance the software's security and only indirectly and unobservable add value to the software (Ng et al., 2009). Fostering and maintaining secure behavior is a major topic in IS (Steinbart et al., 2016; Liang and Xue, 2010), which includes promoting the application of such updates. If updates are rolled out to users, developers of applications or platforms have various options to make them available to users. Updates may be applied consistently or only optional and they may be announced before or after successful implementation. In the near future, they may even be completely

implemented in the background. From a software provider perspective, it thus becomes crucial to understand how their users perceive such distinct update delivery strategies.

Though updates are ubiquitously used and digital businesses heavily depend on their customers' loyalty (i.e., continued use), there is little research on the impact of their delivery strategies on users' beliefs, attitudes, and specifically continuance intentions regarding the updated software (Hong et al., 2011; Claussen et al., 2013). This understanding is essential to fully grasp individual behaviors in digital ecosystems (e.g., Carillo et al., 2014; Liu et al., 2016). Current research often neglects the user perspective and explores software updates mostly from a technical perspective. This includes research on software engineering (Sommerville, 2010), software product lines (Clements and Northrop, 2002), release planning (Svahnberg et al., 2010), and software maintenance (Mens and Demeyer, 2008). Updates may change the software during use and over time, and therefore may have the potential to alter users' beliefs, attitudes, and behaviors in the post-adoption stage (Karahanna et al., 1999; Bhattacharjee, 2001). Increasing the understanding of updates and their delivery strategies from a user's perspective has the potential to significantly increase the body of knowledge of existing post-adoption theory.

However, existing research often tends to conceptualize information systems as monolithic black boxes, rather than as a collection of functionalities and characteristics that are alterable over time (Jaspersen et al., 2005; Benlian, 2015). Moreover, there are several calls for research from IS scholars who criticize the negligence of the IT artifact's role in IS research and suggest focusing on changes in beliefs, attitudes, and behaviors emanating from the IT artifact itself rather than from other IT-unrelated environmental stimuli (Benbasat and Zmud, 2003; Hevner et al., 2004; Orlikowski and Iacono, 2001). Understanding the granularity of software, the changes triggered by updates and the effects of distinct strategies of delivering such updates to users, would help to explain how beliefs, attitudes, and behaviors may fluctuate over time because of the evolving nature of information systems that may be permanently advanced by providers. This study therefore raises the following two research questions:

*RQ1: Does the delivery strategy affect an update's impact on users' continuance intentions?*

*RQ2: Do potential effects of delivery strategies differ between feature and security updates?*

Drawing on the expectation-confirmation theory (Oliver, 1980), that is embedded in the IS continuance model (Bhattacharjee, 2001), we conducted an online experiment with 282 participants to answer these questions. This study thereby contributes to prior research in three important ways. First, we find somewhat different user reactions to major update delivery strategies for security and feature updates. Thereby, we identify update type and notification strategy as crucial moderators for explaining the ongoing use of agile information systems. Our second main contribution is shedding light on the effects of a non-mandatory delivery of updates on the identified effect of updates on users' CI. The finding of a diminished positive effect in the feature update case highlights the pivotal role of ECT and its central effect on IS continuance. Our third and overarching contribution lies in the extension of the predominant view of information systems in post-adoption literature. Here we show how an alterable information system might influence users' attitudes and behaviors during use. Software application developers and platforms may also benefit from this study's results in practice. We find that in most cases, users should be notified of updates (for security updates only after successful implementation), even though a seamless and silent integration of updates has become possible with modern digital ecosystems. Moreover, in situations where the user is involved in accomplishing a task, software providers should avoid rolling out non-mandatory updates. Doing so may wipe out any positive effects and may leave the software in a vulnerable or inferior state.

The remainder of the paper is organized as follows. First, we review relevant literature and develop our hypotheses. We then discuss our research methodology and outline the operationalization of our study. We subsequently present empirical results of our analysis. Finally, we conclude and discuss limitations of this research.

## 2 Theoretical Foundations

### 2.1 Feature Updates and Security Updates

Consistent with previous research (e.g., Dunn, 2004), software updates can be defined as self-contained modules of software that are provided to the user for free, to modify or extend software after it has been rolled out and is already in use. With various terms, software updates have been the subject throughout software engineering literature from a technical perspective (Shirabad et al., 2001; Svahnberg et al., 2010; Weyns et al., 2011). In this context, software release planning refers to the “*idea of selecting the optimum set of features or requirements to deliver in a release within given constraints*” (Svahnberg et al., 2010, p. 1), thus falling within the strategic considerations of a service provider on how and when to deliver which software enhancements to users. In contrast to this rich stream of technical literature dealing with software updates, research on users’ beliefs and attitudes regarding updates has so far been very limited (e.g., Fleischmann et al., 2016). Specifically, essential characteristics of the update’s delivery process such as update notifications or consumption choices in context with different types of updates have so far not been explored.

For this study, we distinguish two basic types of software updates for which user perceptions are quite different (Dinev and Hu, 2007), namely feature and security updates. Feature updates change the core functionality of a software by adding distinct features that are deliberately utilized by users to accomplish the task for which the software is used. In contrast, security updates, falling in the broader category of non-feature updates, do not change the core functionality of software and cannot be directly observed by users, but enhance the software’s protective powers or close vulnerabilities (Ng et al., 2009). Because the user’s interaction with the software may change when the software’s perceived value changes, updates have the potential to influence users’ beliefs, attitudes, and behaviors in the post-adoption stage of IS usage. This may even affect users’ decisions on continued use.

### 2.2 Information Systems Continuance

In the context of post-adoption research (Karahanna et al., 1999; Bhattacharjee, 2001), the term information systems continuance refers to the “*sustained use of an IT by individual users over the long-term after their initial acceptance*” (Bhattacharjee and Barfar, 2011, p. 2). Bhattacharjee (2001) has adopted the expectation-confirmation theory (ECT) (Locke, 1976; Oliver, 1980, 1993; Anderson and Sullivan, 1993) to explore IS users’ intentions to continue or discontinue using an IS. ECT posits, that customers compare their initial expectations with perceived product performance. The discrepancy determines their level of satisfaction. The level of satisfaction further impacts repurchase intention (Oliver, 1980, 1993). Bhattacharjee (2001) has replaced repurchase intention of the ECT model by users’ intention to continue using an IS (CI), suggesting that users compare pre-usage expectations with their experience during IS usage. If perceived performance exceeds (falls short) initial expectations, users experience positive (negative) disconfirmation (DISC), which has a positive impact on their satisfaction (SAT) regarding the IS (Bhattacharjee and Barfar, 2011). Satisfied users intend to continue using the IS, while dissatisfied users discontinue its subsequent use (Oliver, 1980; Bhattacharjee, 2001). Perceived usefulness (PU) captures the expectations about future benefits from IS usage (Bhattacharjee and Barfar, 2011) and has a positive impact on both SAT and on CI (Bhattacharjee, 2001).

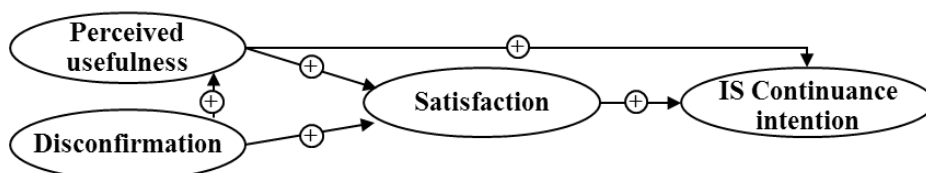


Figure 1: IS Continuance Model (Following Bhattacharjee, 2001)

While the IS continuance model has made valuable contributions to post-adoption research (Bhattacharjee, 2001) it has a static perspective on the IS continuance setting, failing to account for changing user beliefs and attitudes during use. In response to this limitation, several authors have introduced a more dynamic perspective, showing that beliefs and attitudes change from pre-usage to actual usage and during the ongoing usage of an IS (Bhattacharjee and Premkumar, 2004; Kim and Malhotra, 2005; Kim and Son, 2009; Ortiz de Guinea and Markus, 2009; Ortiz de Guinea and Webster, 2013). To investigate this changing nature of the IT artifact and its impact on users' beliefs, attitudes, and behaviors during post-adoption use, we therefore explore software updates and their delivery strategies through the lens of the disconfirmation mechanism in ECT and the IS continuance model.

### 3 Hypotheses Development

In the following section, we will develop our hypotheses on how different update types and delivery strategies in software ecosystems might influence users' post-adoption beliefs and attitudes in non-mandatory or individual use settings. To isolate the core effects, we will focus on a seamless update experience, setting aside notable downsides like download and installation delays. In doing so, we limit ourselves to feasible delivery strategies in modern digital platforms, with either a 'silent' update or a notification given either before or after the update is run. Moreover, we distinguish between the most prevalent and important update types from a user's perspective, those that provide either additional functionality or security enhancements, setting aside minor stability fixes. Finally, to complete our hypotheses, we will posit whether an option to consume an update should be given to the user or not.

#### 3.1 Effects of Notifications for Security Updates

We argue that receiving software updates during post-adoption use can induce positive disconfirmation and increase users' CI (Bhattacharjee, 2001; Hong et al., 2011). According to ECT, the occurrence of positive disconfirmation requires a positive experience compared to prior expectations, i.e. a relative improvement compared to a baseline (Helson, 1964; Oliver, 1980). In the context of software updates, this baseline is formed by the software's pre-update state. An update must therefore exceed this subjective reference point to increase users' CI by leading to a *perceived improvement* of the software (Hong et al., 2011).

Following research on IT security, it is reasonable to assume that users' awareness of the 'protective enhancements' provided by an update plays a major role (i.e. the user has to be aware that something has changed in his favor to feel positively about it). Security updates manifest themselves quite differently than updates providing additional functionality (Ng et al., 2009). They only "*contribute to the wellbeing of their users indirectly and subtly*" (Dinev and Hu, 2007, p. 387). The benefits resulting from security updates cannot be observed directly within the software, as such updates do not add any usable features. Consequently, their value may be derived only from the information provided on the update's intent. A notification of added benefits may therefore substitute the users' experience of an actual change in the software, that the user may otherwise not be aware of (Darby and Karni, 1973).

Such information about an update's intent may be provided to users through notifications either before or after the update is successfully implemented. A notification *before* the execution of the update, however, leaves the user in considerable doubt, as to whether or not the security update was indeed successfully applied (Hoxmeier, 2000; Hong et al., 2011). There is no actual confirming experience on the software's enhancement. Therefore, due to the absent information on successful completion, it will most likely not be perceived as an actual improvement, failing to induce positive disconfirmation and to increase users' CI. In contrast, a notification *after* the successful application of an update clearly conveys the message that the update was completed successfully and that the software therefore, compared to its status quo, may indeed have improved (Hoch and Ha, 1986). Therefore, it is likely that a security update, if announced after successful implementation, will be *perceived* as an *improvement* during use, inducing positive disconfirmation in the sense of ECT (Oliver, 1980). Through an increase

in SAT and PU it will thereby increase users' CI eventually (Bhattacharjee, 2001; Hong et al. 2011). We accordingly derive our first two hypotheses:

*H 1.1: Users who receive a notification before the implementation of a security update will exhibit similar continuance intentions compared to users who did not receive the security update.*

*H 1.2: Users who receive a notification after the successful implementation of a security update will exhibit higher continuance intentions compared to users who did not receive the security update.*

### 3.2 Effects of Notifications for Feature Updates

Moreover, we argue that ECT also applies to the potential effects of feature updates. As reasoned above, to induce positive disconfirmation and to increase CI, an update must lead to a *perceived improvement* of the software. Feature updates can directly contribute to the productivity of the user, and thus elicit a positive experience compared to the software's unupdated state (Hong et al. 2011). However, although feature updates deliver such functionality that directly improves the software with respect to its core purpose, users are often unaware of newly delivered functionality available in the software (Alba and Hutchinson, 1987; Brucks, 1985; Jaspersen et al., 2005; Sun, 2012; Benlian, 2015). The user's capacity of attention is limited, and the user's main task and other interferences will compete for the user's cognitive processing capacity necessary to perceive all available functionality (Kahnemann, 1973; Norman and Bobrow, 1975; Van der Heijden, 1992). Hence, the functionality gains through feature updates may remain unnoticed, if not explicitly presented to users (Sun, 2012).

However, again, the newly available functionality can be made more apparent to users by providing notifications, either before or after the update's successful implementation. In the case of a feature update, though, we posit that the announcement of additional functionality before the update can be confirmed by actual experiences of the specific software enhancements afterwards (Hoxmeier, 2000; Hong et al., 2011). Therefore, notifications before the implementation do not leave users in doubt about the update's success, and thus also have the potential to facilitate a positive experience deriving from the additional new functionality. Summing up, feature updates that are not explicitly announced, may not be recognized by users and therefore may fail to induce positive disconfirmation and eventually increase users' CI. In contrast, feature updates that *are* announced either *before or after* successful implementation will be *perceived* as *improvements* during use, inducing positive disconfirmation in the sense of ECT (Oliver, 1980). Thereby, in this case, through an increase in SAT and PU, users' CI will increase eventually (Bhattacharjee, 2001; Hong et al. 2011). Accordingly, we derive the following three hypotheses:

*H 2.1: Users who receive additional functionality through a feature update without notification will exhibit similar continuance intentions compared to users who did not receive the feature update.*

*H 2.2 Users who receive a notification before the implementation of a feature update will exhibit higher continuance intentions compared to users who did not receive the feature update.*

*H 2.3 Users who receive a notification after the successful implementation of a feature update will exhibit higher continuance intentions compared to users who did not receive the feature update.*

### 3.3 Effects of Non-Mandatory Security and Feature Updates

From conventional practice, one could think that in addition to a pre-update notification, it might be beneficial to provide the option to users on whether to consume an update or not, because doing so would offer more control over the process (Iyengar and Lepper, 2000; Scheibehenne et al., 2010). However, we argue that such a strategy will most likely foster a different result in our case. When users are engaged in using the software to complete a task (Jenkins et al., 2016), the update seems to provide appropriate benefits and, due to our assumption of seamless integration, the update comes with no or only very few downsides, the option to consume an update increases necessary efforts and weakens the potential positive perception of benefits received from an update (Iyengar and Lepper,

2000; Jenkins et al., 2016). Not enhancing the software in the first place, but questioning the update's necessity may leave users in doubt of the update's advantages. As a result, the choice to update may be deferred and the update may therefore fail to exceed prior expectations, as compared to situations where the update is always applied (Jenkins et al., 2016). An obligatory choice can thereby make an update fail to elicit positive disconfirmation through the mechanisms of ECT (Oliver, 1980) and fail to increase users' CI as outlined in our hypothesizing above (Bhattacharjee, 2001; Hong et al. 2011). Summing up, we argue that providing a consumption choice for an update will impair a *perceived improvement* resulting from the functionality gains in cases where the update would otherwise increase users' CI (as argued in hypotheses H2.2). By weakening the update's necessity, a choice to either consume or to dismiss an update will diminish users' potential positive experiences emanating from the update's content. However, in cases where the update does not elicit positive disconfirmation (as argued in our hypotheses H1.2), providing a choice does not harm users' CI. We therefore hypothesize:

*H3.1 Users who have the choice to optionally consume a security update before its conditional implementation will exhibit similar continuance intentions compared to users who consistently receive the update with a notification given beforehand.*

*H3.2 Users who have the choice to optionally consume a feature update before its conditional implementation will exhibit lower continuance intentions compared to users who consistently receive the update with a notification given beforehand.*

## 4 Method

### 4.1 Experimental Design

With the goal to examine the effects of security and feature updates and their delivery strategies on users' CI, we conducted a 2 x 4 between-subjects online-experiment with manipulations of update type (security update vs. feature update) and delivery strategy (no update notification vs. post-update notification vs. pre-update notification vs. pre-update notification and update consumption choice). The design may also be considered as a combined 2 x 3 (update type vs. notification and timing) and 2 x 2 (update type vs. choice) experiment. We carefully developed this design, because an update consumption choice can only be provided by simultaneously notifying users about the upcoming update. However, the chosen design allowed us to both separate the effects of the two factors and to subsequently put them into relation. We opted for an online experiment because it allowed us to investigate and clearly isolate the causal mechanisms that operate between delivery strategies, update types and changes in user attitudes, beliefs, and intentions. We consider this as crucial given that this study is one of the first to explore the effect of different update delivery strategies on users' CI. It also enabled us to account for the claims of numerous researchers to put the IT artifact more at the center of investigation of post-adoption research by using actual changes in an IS as basis for manipulations. The software and the task for which the software had to be used were held constant across all conditions.

The experiment proceeded in four major steps: First, subjects were randomly assigned to one of the eight groups. Second, subjects were instructed to make use of a banking app to check for an outstanding bank transfer (i.e., our cover story) and were then transferred to a fully functional click dummy of a banking app. The app provided an account statement that listed several realistic but random payments, but did not contain the transfer in question. Third, subjects were told, that, on the next day, they would reuse the app to once more check for the outstanding transfer and were then forwarded to the banking app again. In this second usage period the transfer in question was contained towards the end of the list (to equally engage the user in the app). According to the experimental group (see Table 1), for a security update, the app was kept constant in both usage periods (because the security update does not manifest in the user interface) (Group A), only in the second period a 'successfully updated' notification was given (Group B), in the first period an update announcement was given (Group C), and in the first period an update announcement including the option to either dismiss or to install the

update in the background was given (Group D). For a feature update, in the second period a feature was added (Group E), in the second period a feature was added including a ‘successfully updated’ notification (Group F), in the first period and update announcement was given and then a feature was added in the second period (Group G), and in the first period an update announcement including the option to either dismiss or to install the update in the background was given and then, according to the user’s choice, a feature was conditionally added to the app in the second period (Group H). Subsequently, after the two usage periods of the banking app, a post-experimental survey was conducted to assess the subjects’ CI with respect to the software and all further variables (see Measures).

Update type:	Security update			Feature update		
Delivery strategy:		Usage period 1	Usage period 2		Usage period 1	Usage period 2
No notification	<b>A</b>	-	-	<b>E</b>	-	F
Post-notification	<b>B</b>	-	N	<b>F</b>	-	N, F
Pre-notification	<b>C</b>	N	-	<b>G</b>	N	F
Pre-not. and choice	<b>D</b>	N (choice)	-	<b>H</b>	N (choice)	F (conditionally)

Table 1: Experimental design and experimental groups (N: notification, F: feature added).

## 4.2 Manipulation of Independent Variables

To realize our manipulations, we opted for the software context of a banking app running on a mobile application platform to ensure that subjects had previous usage experience and that both security and feature updates would provide relevant value. By choosing a mobile software ecosystem, we could realistically mimic the forthcoming behavior of such platforms (Samat, 2016) and separate the effects of receiving updates from interfering factors like performance or technical issues (Sykes, 2011; Tyre and Orlikowski, 1994). While such downsides have been traditionally associated with updates, however, we argue that modern platforms integrate software updates increasingly frictionless and we are thus confident that we can develop viable implications for many contemporary software ecosystems.

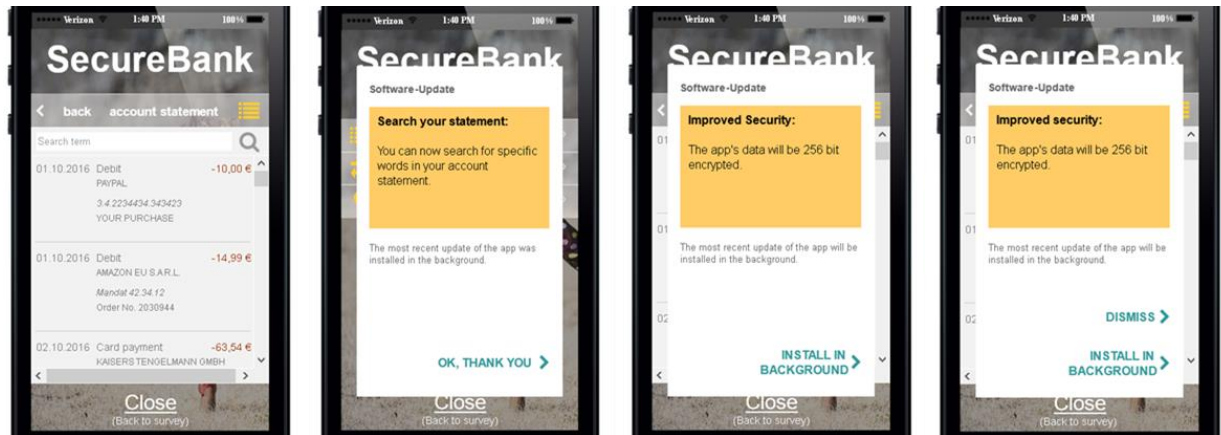


Figure 2. Sample screens of app with no, post-, pre-notification, and additional choice (l.t.r.).

Manipulations of the update type were realized as follows: for the subject of the security and feature update, we first asked 49 participants to rate a list of distinct features of banking apps on perceived importance, which we had compiled through interviews and desk research. Given these insights, we subsequently established the feature ‘search account statement’ and the security enhancement ‘256 Bit encryption’ as subjects for the corresponding feature and security updates. Because enhancements of security do not directly manifest in the user interface other than by a conditional notification (the ex-



perimental group without notification may thereby serve as a control group), only the feature update would also actually add a distinct functionality to the software by providing a search slot above the account statement. Manipulations of the delivery strategy (i.e., notification and conditional choice) were implemented by (1) providing no notification, (2) providing a confirmation layer that describes the successfully installed update and its content after the user re-visits the app, (3) providing an announcement layer that describes the pending update and its content when the user first visits the app (the layer comes up with several seconds delay), and (4) providing the aforementioned layer and additionally giving the option to either accept or to defer the update's installation (see Figure 2).

A qualitative pilot test with five subjects was conducted to ensure that the treatments were manipulated according to the experimental design, that participants would assess the setting as realistic, and that they would understand it well (Perdue and Summers, 1986). Specifically, subjects were asked about the comprehensiveness of the instructions, the effects of the manipulations through the app and the questions in the following questionnaire. In an additional pre-study (n=48), we confirmed the successful manipulation based on measures of control questions. Suggestions were obtained from the participants and the app and the questionnaire were accordingly revised for the main experiment.

### 4.3 Dependent variables, Control Variables and Manipulation Checks

We used validated scales with minor wording changes for all constructs. Measures for CI and DISC were adapted from Bhattacharjee (2001): CI1. *I intend to continue using the app rather than discontinue its use*; CI2. *My intentions are to continue using the app than use any alternative means (traditional banking)*; CI3. *If I could, I would like to discontinue my use of the app* (reverse coded). DISC1. *My experience with using the app was better than what I expected*; DISC2. *The functionality provided by the app was better than what I expected*; DISC3. *Overall, most of my expectations from using the app were confirmed*. Measures for PU and SAT were based on Kim and Son (2009): PU1. *Using the app enhanced my effectiveness in completing the task*; PU2. *Using the app enhanced my productivity in completing the task*; PU3. *Using the app improved my performance in completing the task*. SAT1. *I am content with the features provided by app*; SAT2. *I am satisfied with the features provided by the app*; SAT3. *What I get from using the app meets what I expect for this type of programs*. Because constructs were measured with multiple items, summated scales based on the average scores of the multi-items were used in group comparisons (Zhu et al., 2012). Unless stated otherwise, the questionnaire items were measured on a seven-point-Likert-scale anchored at (1)=strongly disagree and (7)=strongly agree. To ensure successful manipulations we captured whether participants thought that *they had received an update*, what the *subject of the update* was, and if they had been *notified before or after*. Also, in groups with non-mandatory updates, we measure actual *confirmations* and *dismissals* of updates and the participants' *intentions* to *install* or to *not install* such an update. Participants were further asked to what extent they had understood *the items' formulation*, whether they were *able to put themselves in the given situation*, if *the scenario was realistic*, and if they knew *what the goals of the survey were*. The participants' *expertise regarding online banking* was captured on an established four item scale developed by Mishra et al. (1993). We included this control variable as well as the participant's *online banking usage intensity*, *perceived common update frequency*, and finally the subjects' *demographics* (age, gender, profession), to isolate the effects from other possible covariates.

### 4.4 Participants, Incentives and Procedures

Participants were recruited over Clickworker, a German crowdsourcing platform similar to Amazon Mechanical Turk (Paolacci et al., 2010). We offered a small payment for the participation in our online experiment. Overall, 312 subjects started the experiment. The rate of completion was 96%, i.e., a total number of 301 subjects completed the questionnaire. We excluded 19 participants from our final analysis because they did not pass our quality check questions. The average time needed for the completion of the experiment and questionnaire was 9.10 minutes. Of the 282 remaining German speaking participants used in the following analyses, 142 were females and 140 were males. Subjects'

average age was 36.42 ( $\sigma=11.28$ ) years. On average, in one month, 14% of the subjects use online banking up to one time, 20% up to four times, 21% up to eight times, and 45% more than eight times. The average reported expertise with online banking was 5.42 ( $\sigma=1.36$ ) on a seven-point semantic differential scale. More than 40% of the subjects were employees, 21% self-employed, 13% students, and the remainder had various or no occupation. The educational backgrounds of the participants were diverse, including psychology, law, educational sciences, chemistry, computer science, economics, design, agriculture and marketing.

## 5 Data Analysis and Results

### 5.1 Control Variables and Manipulation Check

To confirm a successful randomization, we first searched for differences of the control variables between groups. However, the results of a one-way MANOVA showed no significant differences between groups ( $\lambda=0.83$ ,  $F[49,1365]=1.03$ ,  $p>0.05$ ). Neither of the control variables were significant: age ( $F=1.42$ ,  $df=7$ ,  $p>0.05$ ), gender ( $F=0.61$ ,  $df=7$ ,  $p>0.05$ ), profession ( $F=0.87$ ,  $df=7$ ,  $p>0.05$ ), usage intensity ( $F=1.62$ ,  $df=7$ ,  $p>0.05$ ), update frequency ( $F=1.26$ ,  $df=7$ ,  $p>0.05$ ), and product expertise ( $F=1.39$ ,  $df=7$ ,  $p>0.05$ ). Hence, we concluded that participants' demographics and relevant controls were homogeneous across conditions and did not confound the effects of our manipulations. Finally, we confirmed successful manipulations by performing a Fisher's exact test finding significant differences between conditions in terms of the reported software delivery design type ( $p<0.01$ ) and the reported subject of the update ( $p<0.01$ ). As indicators for the external validity of our findings, we further reviewed participants' answers regarding the realism and adaption of the scenario. For both measures, participants reported high levels on a seven-point-Likert-scale (realism  $\bar{x}=6.32$ ;  $\sigma=1.09$ ; adaption  $\bar{x}=6.35$ ;  $\sigma=1.04$ ). It is therefore reasonable to assume that our manipulations worked as intended, that participants acted typically, and that the setting was realistic.

### 5.2 Measurement Validation

Because we adopted established constructs for our measurement, a confirmatory factor analysis (CFA) was conducted to test the instruments' convergent and discriminant validity (Levine, 2005), using SmartPLS 2 (Chin et al., 2003; Ringle et al., 2005). Table 1 reports the results for the core constructs.

Latent construct	Number of indicators	Range: standardized factor loadings*	Cronbach's alpha	Comp. reliability (pc)	Avg. variance extracted (AVE)
Disconfirmation (DISC)	3	0.869-0.922	0.889	0.931	0.818
Perceived Usefulness (PU)	3	0.930-0.953	0.939	0.961	0.892
Satisfaction (SAT)	3	0.934-0.966	0.948	0.967	0.906
Continuance Intention (CI)	3	0.783-0.936	0.847	0.908	0.768
Note: *All factor loadings are significant at least at the $p<0.01$ level					

Table 2. Results of confirmatory factor analysis for core variables.

All items loaded on the target factors and scored above the threshold of 0.7, indicating proper construct validity (Cook and Campbell, 1979; Bartholomew et al., 2008). AVE values for each construct ranged from 0.768 to 0.906, exceeding the variance due to error (0.5). The constructs were also assessed for reliability using Cronbach's alpha (Cronbach, 1951). A value of at least 0.7 is suggested to indicate adequate reliability which we could confirm for all constructs (Nunnally et al., 1994). Furthermore, the composite reliability of all constructs exceeded 0.7, which is considered the minimum threshold (Hair et al., 2011). Thus, all constructs met the norms for convergent validity. For satisfactory discriminant validity, the square root of the constructs' AVE should be greater than the variance

shared between the constructs in the model (Fornell and Larcker, 1981). All square roots of AVE exceeded inter-construct correlations, indicating proper discriminant validity. Hence, the constructs in our study are theoretically and empirically distinguishable.

### 5.3 Hypotheses Testing

In order to test our hypotheses, we conducted a one-way ANOVA with planned contrast analyses. We found significant differences between groups for DISC ( $F=4.023$ ,  $p<0.01$ ), PU ( $F=3.349$ ,  $p<0.01$ ), SAT ( $F=2.959$ ,  $p<0.01$ ), and CI ( $F=2.511$ ,  $p<0.05$ ). Figure 3 summarizes the results for our main dependent variable CI for both security and feature updates.

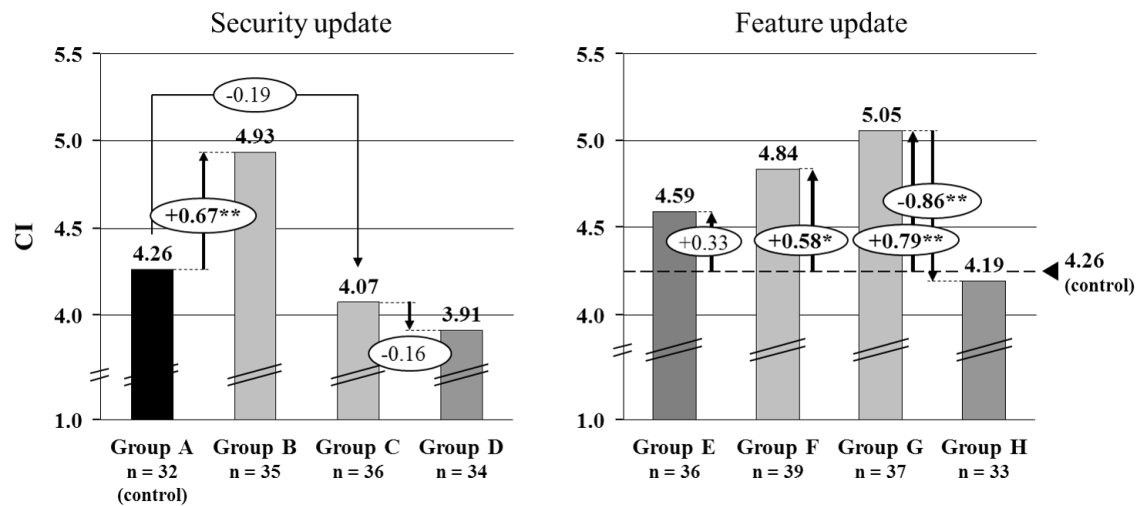


Figure 3. Mean values, differences and significance levels for CI between groups.

Regarding security updates, the contrast analysis revealed that users who received a pre-update notification for the security update showed indifferent reactions in terms of DISC, PU, SAT, and CI compared to users who did not receive any notification (CI:  $\bar{x}$ 's = 4.07 vs. 4.26,  $p>0.1$ ) (see Table 3). This supports our hypothesis 1.1. However, users who received a post-update notification on a security update exhibited significantly higher DISC, PU, SAT, and CI compared to users who did not receive any notification, supporting our hypothesis 1.2 (CI:  $\bar{x}$ 's = 4.93 vs. 4.26,  $p<0.05$ ).

	Security update delivery strategy (n)	DISC	PU	SAT	CI
Groups	A. No notification / control (32)	4.59	5.00	4.88	4.26
	B. Post-notification (35)	5.19	5.63	5.48	4.93
	C. Pre-notification (36)	4.48	4.61	4.78	4.07
	D. Pre-notification and choice (34)	4.59	4.91	5.05	3.91
Diff.	B-A. <sup>1</sup>	<b>0.60**</b>	<b>0.63**</b>	<b>0.60**</b>	<b>0.67**</b>
	C-A. <sup>1</sup>	-0.11	-0.39	-0.10	-0.19
	D-C. <sup>1</sup>	0.11	0.30	0.27	-0.16

Table 3. Mean values, differences and significance levels for security update groups.

<sup>1</sup> ANOVA-tests with planned contrast analyses; \*\*\*  $p<0.01$ , \*\*  $p<0.05$ , \*  $p<0.1$  (one-sided);

Investigating feature updates, the contrast analysis revealed that users who received the update but were not notified at all showed *indifferent* reactions in terms of DISC, PU, SAT, and CI compared to users who did not receive any update (CI:  $\bar{x}$ 's = 4.59 vs. 4.26,  $p > 0.1$ ) (see Table 4). This supports our hypothesis 2.1. However, users who received a pre-update notification on the feature update exhibited significantly *higher* DISC, PU, SAT, and CI compared to users who did not receive any update, supporting our hypothesis 2.2 (CI:  $\bar{x}$ 's = 5.05 vs. 4.26,  $p < 0.05$ ). Likewise, users provided with a post-update notification exhibited significantly *higher* DISC, PU, SAT, and CI compared to users who did not receive any update, which supports our hypothesis 2.3 (CI:  $\bar{x}$ 's = 4.84 vs. 4.26,  $p < 0.1$ ).

	Feature update delivery strategy (n)	DISC	PU	SAT	CI
Groups	E. No notification (36)	4.84	4.78	5.05	4.59
	F. Post-notification (39)	5.61	5.50	5.74	4.84
	G. Pre-notification (37)	5.62	5.68	5.79	5.05
	H. Pre-notification and choice (33)	4.65	4.91	4.91	4.19
Diff.	E-A. <sup>1</sup>	0.25	-0.22	0.17	0.33
	F-A. <sup>1</sup>	<b>1.01***</b>	<b>0.50*</b>	<b>0.89***</b>	<b>0.58*</b>
	G-A. <sup>1</sup>	<b>1.03***</b>	<b>0.68**</b>	<b>0.92***</b>	<b>0.79**</b>
	H-G. <sup>1</sup>	<b>-0.98***</b>	<b>-0.78***</b>	<b>-0.88***</b>	<b>-0.86**</b>

Table 4. Mean values, differences and significance levels for feature update groups.

Finally, regarding non-mandatory updates, the results of the contrast analysis revealed that for security update there was *no significant difference* in terms of DISC, PU, SAT, and CI (Group D-C) between users who had the choice to either consume the security update or not, compared to users who received the security update in any case (CI:  $\bar{x}$ 's = 3.91 vs. 4.07,  $p > 0.1$ ). This supports our hypothesis H 3.1. On the contrary, users who had the choice to either consume the feature update or not (Group H-G) exhibited significantly *lower* DISC, PU, SAT, and CI, compared to users who received the feature update in any case, as predicted by our hypothesis H 3.2. Further inspecting the actual decisions of update installations, based on a chi-square test, we could not find a significant difference between security updates (Confirmed vs. dismissed: 11 vs. 23) and feature updates (Confirmed vs. dismissed: 10 vs. 23) ( $\chi^2=0.033$ ,  $p > 0.1$ ). However, in the reported intentions to dismiss or confirm such an update, we could find a difference between security updates (Confirmed vs. dismissed: 65 vs. 1) and feature updates (Confirmed vs. dismissed: 52 vs. 14) ( $\chi^2=12.711$ ,  $p < 0.001$ ).

## 6 Discussion

This study sought to achieve two main objectives: (1) to examine the effects of different software update delivery strategies on users' continuance intentions, and (2) to investigate potential distinctions between the natures of security and feature updates. To achieve these two objectives, we drew on the IS continuance model and we investigated our hypotheses based on an online-experiment with 282 participants in the context of a banking app, operated on a mobile platform.

Our results reveal that users who receive a security update show divergent reactions to being notified of the update before or after its successful implementation. In the case of a post-update notification (Group B), users showed a significantly higher CI. This finding strengthens the notion that for security updates a notification on the update's successful implementation may serve as a proxy for its actual realization (which is not observable from a user's perspective). However, in the case of an ex-ante notification (Group C), no significant change in CI could be observed. Given our first finding, this may seem somewhat counter-intuitive at first. However, it may be explained by the fact that the results of a security update are not physically observable in the software (Ng et al., 2009). Thus, users are being left in vagueness about the update's actual implementation. Regarding feature updates, users receiving additional new functionality without further notification (Group E), did not show a significant increase

in CI, despite this increased value provided by the software. This somewhat unexpected result may be explained by the users' attention bound to the task users had to accomplish (Kahnemann, 1973), leaving the additional functionality unnoticed. Only in both cases when the feature update was announced before or after successful implementation (Group F and G), we found a significant increase in users' CI. In those cases, the noticeable 'gift' of additional functionality was then able to elicit positive disconfirmation, thereby increasing users' CI.

In addition, we could evidence that updates that are delivered with a non-mandatory strategy do not increase users' CI. In case of a security update (Group D), providing the update to users as an optional alternative did not increase users' CI, compared to the ex-ante announcement and a mandatory rollout. In case of a feature update (Group H), a consumption choice even perhaps significantly decreased users' CI compared to an ex-ante announcement and a mandatory installation. Probably by questioning the necessity of an update and thereby preventing the consumption in many cases, such an option inhibited a potential positive experience. Inspecting the numbers of actual confirmations and dismissals for both update types, surprisingly, we could observe that they were more often dismissed than consumed with a rate that did not differ significantly between the two types. On the contrary, the intention of users to install security updates was significantly higher than for feature updates, which stresses the users' perception of importance of security updates. This finding again highlights a gap between intentions and actual behavior and thereby provides avenues for further research (Jenkins et al., 2016).

## 6.1 Implications for Research

The paper makes three main contributions to the literature. First, we identify update type and delivery strategy as crucial moderators for the positive effect of an update on users' CI. We find that providing a security update increases users' CI by disconfirming previous expectations only if it is announced after successful implementation. A feature update, on the other hand, induces a positive reaction in all situations in which it is announced in addition to its rollout (i.e., before or after implementation), while it does not have such a potential if it is silently implemented in the background. This interaction emphasizes the importance of a joint consideration of the IT artifacts' and the update's characteristics when investigating user behavior. Our second main contribution is shedding light on the effects of a non-mandatory update on the identified effect of updates on users' CI. Specifically, we find that a positive effect of feature updates on CI, by positively disconfirming previous expectations, is diminished when the update is provided only optionally. Nevertheless, CI remains unaffected for security updates in this case. These findings once again highlight the pivotal role of ECT and its central effect on IS continuance compared to other factors. Both findings add to the body of knowledge on software updates. Our third and overarching contribution lies in showing how a malleable information system might influence users' attitudes and behaviors during post-adoption use. We answer the calls of several IS researchers by extending the still predominant view of post-adoption literature on the IT artifact as a monolithic block to a more flexible perspective that considers information systems as a modular composition of functionality that may change over time (Jasperson et al., 2005; Benbasat and Barki, 2007; etc.). We complement existing IS post-adoption literature and research on digital ecosystems (Carillo et al., 2014; Liu et al., 2016) through the notion that users' beliefs and attitudes might change with the advancement of the system.

## 6.2 Implications for Practice

Our results have important and viable implications for practice, particularly for contemporary software ecosystem settings, where updates are integrated increasingly frictionless. First, despite the extensive use of updates by organizations to enhance and progress their services on digital platforms, it is surprising that insights on how these updates and their delivery are perceived by users are still scarce. This leaves practitioners without guidance. From the results of our experimental study we can conclude that developers of applications and platforms should rather announce feature and security enhancements instead of implementing them silently. However, for security enhancements, the only

helpful measure for developers in terms of the user's loyalty (i.e., CI) is to announce such updates only after the successful implementation. More specifically, our findings suggest, that only in cases when the user is notified after successful implementation of a security update, it has the potential to increase users' CI above and beyond a level generated by software where the security update was communicated before implementation or not communicated at all. With respect to feature updates, developers can learn from this study's results that they can increase their users' loyalty by announcing them before or after successful implementation. Both strategies should be preferred over not at all communicating such enhancements, as updates won't be always noticed by users in the software itself.

Finally, it is not advisable for developers of applications and platforms to provide users the option to either consume or to defer an update. It is better to apply updates consistently. Providing such an option may not only diminish an update's positive effect, but may leave the software in an inferior state. In today's interconnected and quickly changing multi-device and multi-platform environments, users heavily rely on security and on a comparable feature set with respect to competitors' solutions. To avoid losing customers from vulnerabilities or major disadvantages (even if only temporary), platform and application providers should thus quickly respond to such needs and roll out according changes consistently. It should be noted, however, that these findings only apply to situations where the update process does not come with major downsides and the update's contents are unquestionably helpful.

### **6.3 Conclusion, Limitations, and Future Research**

In modern digital ecosystems, software updates have become a pervasively used instrument for businesses to enhance their digital services over time. Despite this prevalence, the effects of update delivery strategies on crucial post-adoption user reactions have remained largely unexplored. This study's diverse findings highlight the importance of a profound understanding of update delivery strategies in evolving software ecosystems for both researchers and practitioners. Security updates have the potential to increase users' CI only if they are communicated after implementation, while feature updates have such a potential if at least communicated at any time. Providing an option to defer an update however seems to be unfavorable or even harmful, as it may diminish any positive effects elicited by an update, and in the end, because users tend to dismiss them considerably, may leave the application in an inferior or even vulnerable state.

Three limitations of this study are noteworthy and provide avenues for future research. First, in our experiment, we utilized a self-developed, simplified click dummy of a banking app with a homogeneous feature set. This quasi-realistic setting of a digital ecosystem's software required subjects to adapt to the software and setting. Hence, we controlled for adaption and perceived realism of the scenario. Based on the convincing results for these controls, we are confident that our study's implications are applicable to real usage settings. Nonetheless, future studies could investigate actual usage experiences with real software to validate our findings. Second, we identified security and feature updates in the banking context as crucial update types for examining the effects of update delivery strategies on users' CI. Also, subjects were recruited in Germany. Since security plays a major role in the banking context and attitudes towards security might differ between countries, future studies are encouraged to validate our findings in different contexts and cultural settings. Furthermore, complementary qualitative studies (e.g., thought-listing) could substantiate our theoretical reasoning and could uncover additional mediating mechanisms. Finally, we conducted a controlled experiment with the purpose of obtaining results with a high internal validity. This required some reasonable but strict assumptions, such as a limited observation period, an identical and linear course of events, a determined task and ex-post measurement of variables. Future studies are encouraged to complement our findings by conducting longitudinal field experiments to advance the external validity of our findings over longer timespans and to account for learning effects. Also, settings with repeated updates with participants' evaluations measured at several points in time could provide additional evidence for the robustness of our findings. In the further course, research should seek to deepen the understanding of how dynamic software ecosystems need to be shaped to both satisfy and protect users by considering individual behaviors.

## References

- Alba, J. W. and J. W. Hutchinson (1987). "Dimensions of Consumer Expertise." *The Journal of Consumer Research* 13 (4), 411-454.
- Anderson, E. W. and M. W. Sullivan (1993). "The Antecedents and Consequences of Customer Satisfaction for Firms." *Marketing Science* 12 (2), 125-143.
- Bartholomew, D. J., F. Steele, J. Galbraith and I. Moustaki (2008). *Analysis of Multivariate Social Science Data*. 2nd Edition. London: Chapman and Hall/CRC.
- Benbasat, I. and H. Barki (2007). "Quo Vadis, TAM?" *Journal of the Association for Information Systems* 8 (4), 211-218.
- Benbasat, I. and R. W. Zmud (2003). "The Identity Crisis within the IS Discipline: Defining and Communicating the Discipline's Core Properties." *MIS Quarterly* 27 (2), 183-194.
- Benlian, A. (2015). "IT Feature Use over Time and its Impact on Individual Task Performance." *Journal of the Association for Information Systems* 16 (3), 144-173.
- Bhattacharjee, A. (2001). "Understanding Information Systems Continuance: An Expectation-Confirmation Model." *MIS Quarterly* 25 (3), 351-370.
- Bhattacharjee, A. and A. Barfar (2011). "Information Technology Continuance Research: Current State and Future Directions." *Asia Pacific Journal of Information Systems* 21 (2), 1-18.
- Bhattacharjee, A. and G. Premkumar (2004). "Understanding Changes in Belief and Attitude toward Information Technology Usage: A Theoretical Model and Longitudinal Test." *MIS Quarterly* 28 (2), 229-254.
- Brucks, M. (1985). "The Effects of Product Class Knowledge on Information Search Behavior." *Journal of Consumer Research*. 12 (2), 1-16.
- Carillo, K. D. A., E. Scornavacca and S. Za (2014). "An Investigation of the Role of Dependency in Predicting Continuance Intention to Use Ubiquitous Media Systems: Combining a Media System Perspective with Expectation-Confirmation Theories." *ECIS 2014 Proceedings*. Münster: Germany.
- Chin, W. W., B. L. Marcolin and P. R. Newsted (2003). "A Partial Least Squares Latent Variable Modeling Approach for Measuring Interaction Effects: Results from a Monte Carlo Simulation Study and an Electronic-Mail Emotion/Adoption Study." *Information Systems Research* 14 (2), 189-217.
- Claussen, J., T. Kretschmer and P. Mayrhofer (2013). "The Effects of Rewarding User Engagement: The Case of Facebook Apps." *Information Systems Research* 24 (1), 186-200.
- Clements, P. and L. Northrop (2002). *Software Product Lines: Practices and Patterns*. 3rd Edition. Boston, MA: Addison-Wesley Reading.
- Cook, T. D. and D. T. Campbell (1979). *Quasi-Experimentation: Design & Analysis Issues for Field Settings*. Boston: Houghton Mifflin.
- Cronbach, L. J. (1951). "Coefficient Alpha and the Internal Structure of Tests." *Psychometrika* 16 (3), 297-334.
- Darby, M. R. and E. Karni (1973). "Free Competition and the Optimal Amount of Fraud." *Journal of Law and Economics* 16 (1), 67-88.
- Dinev, T. and Q. Hu (2007). "The Centrality of Awareness in the Formation of User Behavioral Intention toward Protective Information Technologies." *Journal of the Association for Information Systems* 8 (7), 386-408.
- Dunn, K. (2004). "Automatic Updates: Automatic Update Risks: Can Patching Let a Hacker in?" *Network Security* 2004 (7), 5-8.
- Fleischmann, M., M. Amirpur, T. Grupp, A. Benlian and T. Hess (2016). "The Role of Software Updates in Information Systems Continuance – An Experimental Study from a User Perspective." *Decision Support Systems* (83), 83-96.

- Fornell, C. and D. F. Larcker (1981). "Evaluating Structural Equation Models with Unobservable Variables and Measurement Error." *Journal of Marketing Research* 18 (1), 39-50.
- Hair, J. F., C. M. Ringle and M. Sarstedt (2011). "PLS-SEM: Indeed a Silver Bullet." *Journal of Marketing Theory and Practice* 19 (2), 139-151.
- Helson, H. (1964). *Adaptation-Level Theory: An Experimental and Systematic Approach to Behavior*. New York: Harper and Row.
- Hevner, A. R., S. T. March, J. Park and S. Ram (2004). "Design Science in Information Systems Research." *MIS Quarterly* 28 (1), 75-105.
- Hoch, S. J. and Y.-W. Ha (1986). "Consumer Learning: Advertising and the Ambiguity of Product Experience." *Journal of Consumer Research* 13 (3), 221-233.
- Hong, W., J. Y. Thong, L. C. Chasalow and G. Dhillon (2011). "User Acceptance of Agile Information Systems: A Model and Empirical Test." *Journal of Management Information Systems* 28 (1), 235-272.
- Hoxmeier, J. A. (2000). "Software Preannouncements and Their Impact on Customers' Perceptions and Vendor Reputation." *Journal of Management Information Systems* 17 (1), 115-139.
- Iyengar, S. S. and M. R. Lepper (2000). "When Choice is Demotivating: Can One Desire Too Much of a Good Thing?" *Journal of Personality and Social Psychology* 79 (6), 995-1006.
- Jasperson, J. S., P. E. Carter and R. W. Zmud (2005). "A Comprehensive Conceptualization of Post-Adoptive Behaviors Associated with Information Technology Enabled Work Systems." *MIS Quarterly* 29 (3), 525-557.
- Jenkins, L., B. B. Anderson, A. Vance, C. B. Kirwan and D. Eargle (2016). "More Harm Than Good? How Messages That Interrupt Can Make Us Vulnerable." *Information Systems Research* 27 (4), 880-896.
- Kahnemann, D. (1973). *Attention and effort*. Englewood Cliffs: Prentice-Hall.
- Karahanna, E., D. W. Straub and N. L. Chervany (1999). "Information Technology Adoption Across Time: A Cross-Sectional Comparison of Pre-Adoption and Post-Adoption beliefs." *MIS Quarterly* 23 (2), 183-213.
- Kim, S. S. and J. Y. Son (2009). "Out of Dedication or Constraint? A Dual Model of Post-Adoption Phenomena and Its Empirical Test in the Context of Online Services." *MIS Quarterly* 33 (1), 49-70.
- Kim, S. S. and N. K. Malhotra (2005). "A Longitudinal Model of Continued Is Use: An Integrative View of Four Mechanisms Underlying Postadoption Phenomena." *Management Science* 51 (5), 741-755.
- Levine, T. R. (2005). "Confirmatory Factor Analysis and Scale Validation in Communication Research." *Communication Research Reports* 22 (4), 335-338.
- Liang, H. and Y. Xue (2010). "Understanding Security Behaviors in Personal Computer Usage: A Threat Avoidance Perspective." *Journal of the Association for Information Systems* 11 (7), 394-413.
- Liu, J., V. Abhishek and B. Li (2016). "The Impact of Mobile Adoption on Customer Omni-Channel Banking Behavior." *ICIS 2016 Proceedings*. Dublin: Ireland.
- Locke, E. A. (1976). "The Nature and Causes of Job Satisfaction." In: *Handbook of Industrial and Organizational Psychology*. Ed. by M.D. Dunnette. Chicago: Rand McNally College Pub. Co., 1297-1349.
- Mens, T. and S. Demeyer (2008). *Software Evolution*. Berlin and Heidelberg: Springer.
- Mishra, S., U. N. Umesh and Jr. D. E. Stem (1993). "Antecedents of the Attraction Effect: An Information-Processing Approach." *Journal of Marketing Research* 30 (3), 331-349.
- Ng, B.-Y., A. Kankanhalli, and Y. Xu (2009). "Studying users' computer security behavior: A health belief perspective." *Decision Support Systems* 2009 (46), 815-825.
- Norman, D. A. and D. G. Bobrow (1975). "On Data-limited and Resource-limited Process." *Cognitive Psychology* 7, 20.
- Nunnally, J. C. (1994). *Psychometric Theory*. 3rd Edition. New York: McGraw-Hill.



- Oliver, R. L. (1980). "A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions." *Journal of Marketing Research* 17 (4), 460-469.
- Oliver, R. L. (1993). "Cognitive, Affective, and Attribute Bases of the Satisfaction Response." *Journal of Consumer Research* 20 (3), 418-430.
- Orlikowski, W. J. and C. S. Iacono (2001). "Research Commentary: Desperately Seeking the "It" in It Research—a Call to Theorizing the It Artifact." *Information Systems Research* 12 (2), 121-134.
- Ortiz de Guinea, A. and M. L. Markus (2009). "Why Break the Habit of a Lifetime? Rethinking the Roles of Intention, Habit, and Emotion in Continuing Information Technology Use." *MIS Quarterly* 33 (3), 433-444.
- Ortiz de Guinea, A., and J. Webster (2013). "An Investigation of Information Systems Use Patterns: Technological Events as Triggers, the Effect of Time, and Consequences for Performance." *MIS Quarterly* 37 (4), 1165-1188.
- Paolacci, G., J. Chandler and P.G. Ipeirotis (2010). "Running Experiments on Amazon Mechanical Turk." *Judgment and Decision Making* 5 (5), 411-419.
- Perdue, B. C. and J. O. Summers (1986). "Checking the Success of Manipulations in Marketing Experiments." *Journal of Marketing Research* 23 (4), 317-326.
- Ringle, C. M., S. Wende and A. Will (2005). *Smart PLS 2.0 (M3)*. Hamburg: University of Hamburg. URL: <http://www.smartpls.de>.
- Samat, S. (2016). *Android 7.0 Nougat: a more powerful OS, made for you*. URL: <https://blog.google/products/android/android-70-nougat-more-powerful-os-made/> (visited on 11/14/2016).
- Scheibehenne, B., R. Greifeneder and P. M. Todd (2010). "Can There Ever Be Too Many Options? A Meta-Analytic Review of Choice Overload." *Journal of Consumer Research* 37 (3), 409-425.
- Shirabad, J. S., T. C. Lethbridge and S. Matwin (2001). "Supporting Software Maintenance by Mining Software Update Records." *Proceedings of the International Conference on Software Maintenance*. Florence, Italy, 22-31.
- Sommerville, I. (2010). *Software Engineering: International Version*. 9th Edition. Boston, MA: Pearson.
- Steinbart, P. J., M. J. Keith and J. Babb (2016). "Examining the Continuance of Secure Behavior: A Longitudinal Field Study of Mobile Device Authentication." *Information Systems Research* 27 (2), 219-239.
- Sun, S. (2012). "Understanding User Revisions When Using Information System Features: Adaptive System Use and Triggers." *MIS Quarterly* 36 (2), 453-478.
- Svahnberg, M., T. Gorschek, R. Feldt, R. Torkar, S. B. Saleem and M. U. Shafique (2010). "A Systematic Review on Strategic Release Planning Models." *Information and Software Technology* 52 (3), 237-248.
- Sykes, E. R. (2011). "Interruptions in the workplace: A case study to reduce their effects." *International Journal of Information Management* 31 (4), 385-394.
- Tyre, M. J. and W. J. Orlikowski (1994). "Windows of Opportunity: Temporal Patterns of Technological Adaptation in Organizations." *Organization Science* 5 (1), 98-118.
- Van der Heijden, A. H. C. (1992). *Selective Attention in Vision*. New York: Routledge.
- Weyns, D., B. Michalik, A. Helleboogh and N. Boucke (2011). "An Architectural Approach to Support Online Updates of Software Product Lines." *Proceedings of the 9th Working IEEE/IFIP Conference on Software Architecture*. Boulder, CO, 204-213.
- Zhu, M., D. M. Billeter and J. J. Inman (2012). "The Double-Edged Sword of Signaling Effectiveness: When Salient Cues Curb Postpurchase Consumption." *Journal of Marketing Research* 49 (1), 26-38.